

**2025**  
**CURSO DE**  
**ESTADÍSTICA**  
**ENZOEM**

UNIDAD DE INVESTIGACIÓN COMPETITIVA DE ZOONOSIS Y ENFERMEDADES EMERGENTES



CAMPUS DE  
RABANALES

6, 8  
MAYO

11, 13, 16  
JUNIO

2, 4  
SEPTIEMBRE

[WWW.CURSOENZOEM.ESMEETINGEVENTOS.ES](http://WWW.CURSOENZOEM.ESMEETINGEVENTOS.ES)

José Antonio Blanco-Aguiar (IREC-JCCM,CSIC,UCLM)  
Javier Fernández-López (IREC-JCCM,CSIC,UCLM)

Comisión de Formación y Comisión Científica  
Unidad de Investigación Competitiva ENZOEM  
Universidad de Córdoba



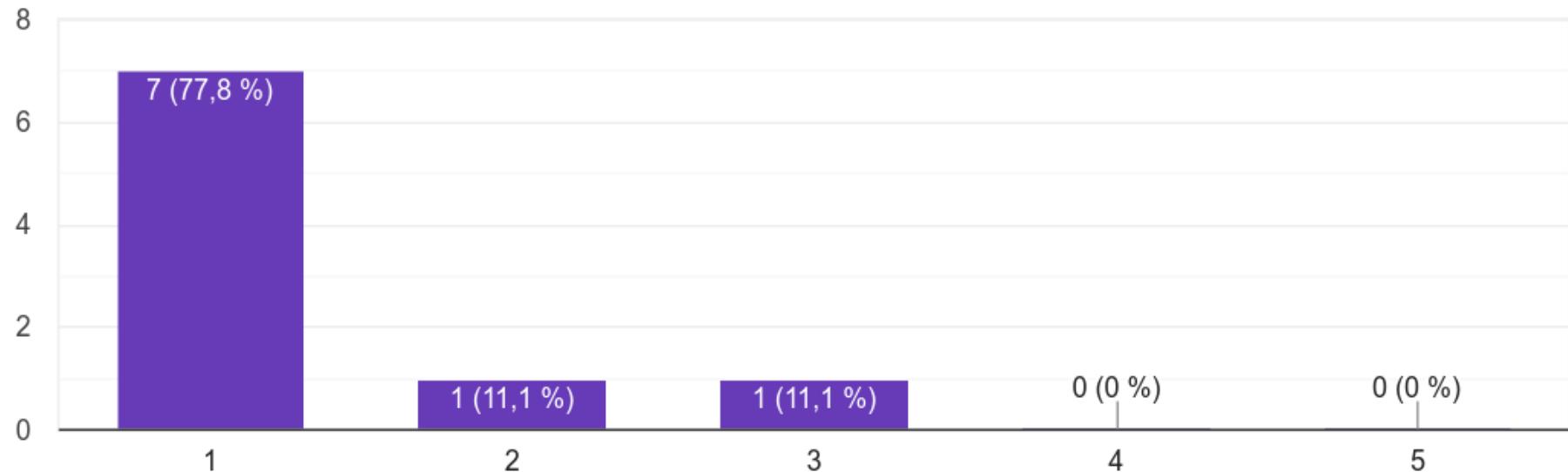
UNIVERSIDAD  
DE  
CÓRDOBA

**Módulo 1. Introducción a R y estadística básica**

# Sobre vosotr@s...

¿Haces uso del lenguaje R?

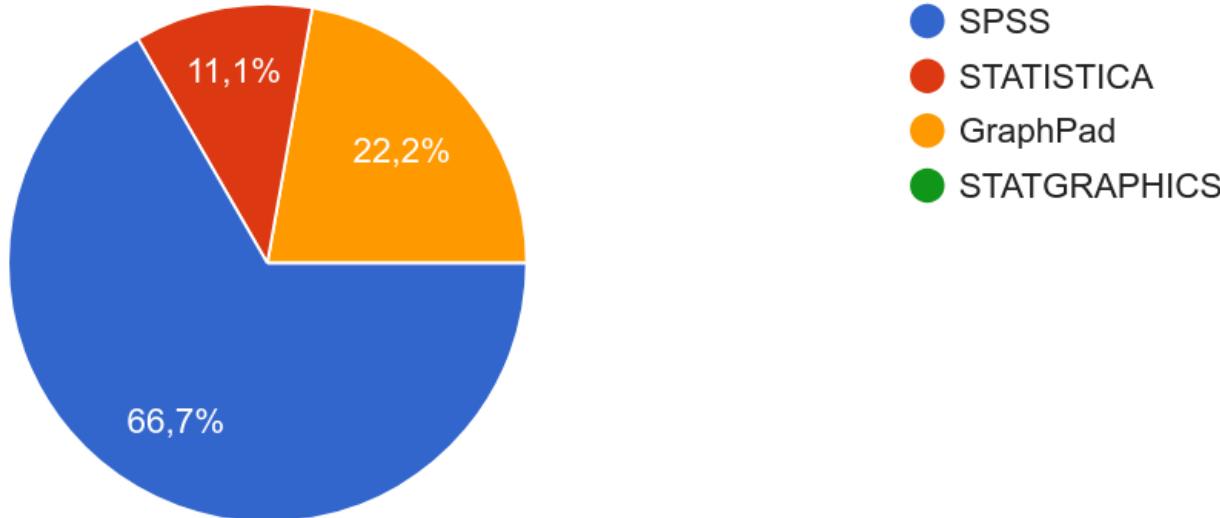
9 respuestas



# Sobre vosotr@s...

¿Qué otros softwares estadísticos has utilizado?

9 respuestas



# Sobre vosotr@s...

¿Qué tipo de test estadísticos sueles utilizar en tu trabajo?

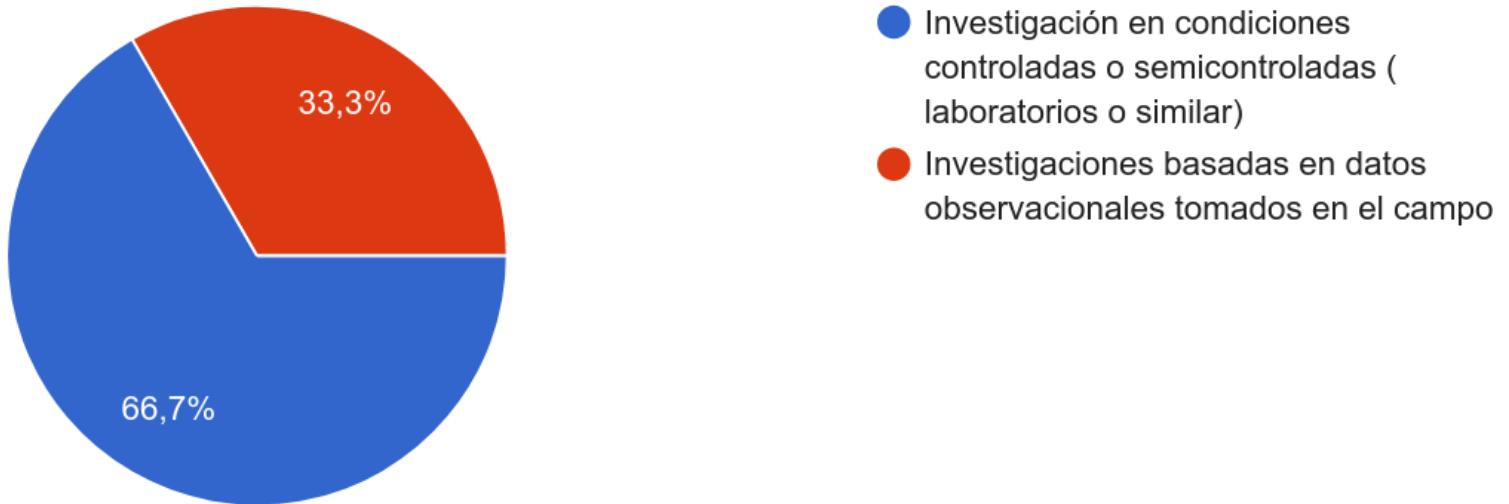
9 respuestas



# Sobre vosotr@s...

¿Qué tipo de investigación sueles desarrollar habitualmente?

9 respuestas



# ¿Qué se espera del curso?

- Imposible en 8 horas (aprendizaje por *sedimentación*)
- Conocer el mecanismo básico y los fundamentos
- Tener documentación y materiales
- “Hacerlo nuestro” cuanto antes

# Módulo 1. Introducción a R y estadística básica

1. El lenguaje R y el entorno R-Studio: funciones y objetos
2. Flujo de trabajo en R: directorio de trabajo, lectura y escritura de datos
3. Visualización de datos en R
4. Distribuciones de probabilidad y estadística básica

**Web dedicada:** [https://jabiologo.github.io/web/tutorials/enzoem\\_1\\_1.html](https://jabiologo.github.io/web/tutorials/enzoem_1_1.html)

# Web dedicada: [https://jabiologo.github.io/web/tutorials/enzoem\\_1\\_1.html](https://jabiologo.github.io/web/tutorials/enzoem_1_1.html)

## Presentación

1. El lenguaje R y el entorno R-Studio: funciones y objetos
2. Flujo de trabajo en R: directorio de trabajo, lectura y escritura de datos
3. Visualización de datos en R y ggplot2
4. Distribuciones de probabilidad y estadística básica

ejemplo, el paquete `lme4` se utiliza frecuentemente para ajustar modelos generales lineales mixtos. Podemos descargarlo, instalarlo y cargarlo en nuestra sesión de R de la siguiente manera:

```
# Descargamos e instalamos el paquete  
install.packages("lme4")  
# Cargamos el paquete en nuestra sesión  
library(lme4)
```

Una de las grandes ventajas (o inconvenientes?) de R es que es un software libre, por lo que cualquiera puede desarrollar sus propios paquetes con las herramientas (funciones) que necesite y ponerlo a disposición de la comunidad de usuarios. Si tenéis curiosidad, [aquí](#) podéis encontrar un pequeño tutorial sobre como hacerlo.

## Los objetos

Los **objetos** en R son los contenedores donde almacenamos los resultados (outputs) de las funciones. Podemos identificarlos porque suelen aparecer por primera vez precediendo a los caracteres `<-`, que simbolizan una flecha que señala hacia la izquierda. Cada vez que se quiera crear un objeto se le ha de dar un nombre, el que queramos, aunque suele ser conveniente darle un nombre que tenga sentido. Por ejemplo, vamos a almacenar en un objeto que vamos a llamar "numeros" la concatenación de valores que creamos anteriormente:

```
# Almacenamos en un objeto llamado "numeros" el resultado de concatenar 3, 7, 12 y 4  
numeros <- c(3, 7, 12, 4)
```

```
# Ahora podemos "llamar" a "numeros" para ver qué tiene dentro  
numeros
```

```
## [1] 3 7 12 4
```

# ¿Por qué R?

- **R es una herramienta libre y gratuita para todos los sistemas operativos.**
- **R no es solamente un software para análisis estadístico.**
- **Es muy probable que tarde o temprano tengáis que utilizar esta herramienta.**



```
javifl@javifl-NL40-50CU:~$ R
R version 4.4.1 (2024-06-14) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu

R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

[El espacio de trabajo previamente guardado ha sido restaurado]
> 
```

# Diferencia entre R y RStudio

- R ~ “motor”
- RStudio ~ chasis/carcasa



The image shows a screenshot of the RStudio IDE interface. The interface is divided into four main panels, each highlighted with a red border:

- SCRIPT**: The top-left panel contains a code editor window titled "Untitled1\*". It displays the first line of an R script: "1 # Este es un nuevo script!".
- ENTORNO**: The top-right panel is the "Environment" tab, which shows the global environment. It displays the message "Environment is empty".
- CONSOLA**: The bottom-left panel is the "Console" tab, showing the R startup message and license information. The message includes details about R version 4.4.1, the platform (x86\_64-pc-linux-gnu), and the GPL license.
- PLOTS/AYUDA...**: The bottom-right panel is the "Plots" tab, which is currently empty.

# Web dedicada: [https://jabiologo.github.io/web/tutorials/enzoem\\_1\\_1.html](https://jabiologo.github.io/web/tutorials/enzoem_1_1.html)

## Presentación

1. El lenguaje R y el entorno R-Studio: funciones y objetos
2. Flujo de trabajo en R: directorio de trabajo, lectura y escritura de datos
3. Visualización de datos en R y ggplot2
4. Distribuciones de probabilidad y estadística básica

ejemplo, el paquete `lme4` se utiliza frecuentemente para ajustar modelos generales lineales mixtos. Podemos descargarlo, instalarlo y cargarlo en nuestra sesión de R de la siguiente manera:

```
# Descargamos e instalamos el paquete  
install.packages("lme4")  
# Cargamos el paquete en nuestra sesión  
library(lme4)
```

Una de las grandes ventajas (o inconvenientes?) de R es que es un software libre, por lo que cualquiera puede desarrollar sus propios paquetes con las herramientas (funciones) que necesite y ponerlo a disposición de la comunidad de usuarios. Si tenéis curiosidad, [aquí](#) podéis encontrar un pequeño tutorial sobre como hacerlo.

## Los objetos

Los **objetos** en R son los contenedores donde almacenamos los resultados (outputs) de las funciones. Podemos identificarlos porque suelen aparecer por primera vez precediendo a los caracteres `<-`, que simbolizan una flecha que señala hacia la izquierda. Cada vez que se quiera crear un objeto se le ha de dar un nombre, el que queramos, aunque suele ser conveniente darle un nombre que tenga sentido. Por ejemplo, vamos a almacenar en un objeto que vamos a llamar "numeros" la concatenación de valores que creamos anteriormente:

```
# Almacenamos en un objeto llamado "numeros" el resultado de concatenar 3, 7, 12 y 4  
numeros <- c(3, 7, 12, 4)
```

```
# Ahora podemos "llamar" a "numeros" para ver qué tiene dentro  
numeros
```

```
## [1] 3 7 12 4
```

# Prácticas 1.1 y 1.2: Flujo de trabajo en R y RStudio

# Prácticas 1.1 y 1.2: Flujo de trabajo en R y RStudio



# Prácticas 1.1 y 1.2: Flujo de trabajo en R y RStudio

Ziemann et al. *Genome Biology* (2016) 17:177  
DOI 10.1186/s13059-016-1044-7

Genome Biology

COMMENT

Open Access



CrossMark

## Gene name errors are widespread in the scientific literature

Mark Ziemann<sup>1</sup>, Yotam Eren<sup>1,2</sup> and Assam El-Osta<sup>1,3\*</sup>

### Abstract

The spreadsheet software Microsoft Excel, when used with default settings, is known to convert gene names to dates and floating-point numbers. A programmatic scan of leading genomics journals reveals that approximately one-fifth of papers with supplementary Excel gene lists contain erroneous gene name conversions.

**Keywords:** Microsoft Excel, Gene symbol, Supplementary data

**Abbreviations:** GEO, Gene Expression Omnibus; JIF, journal impact factor

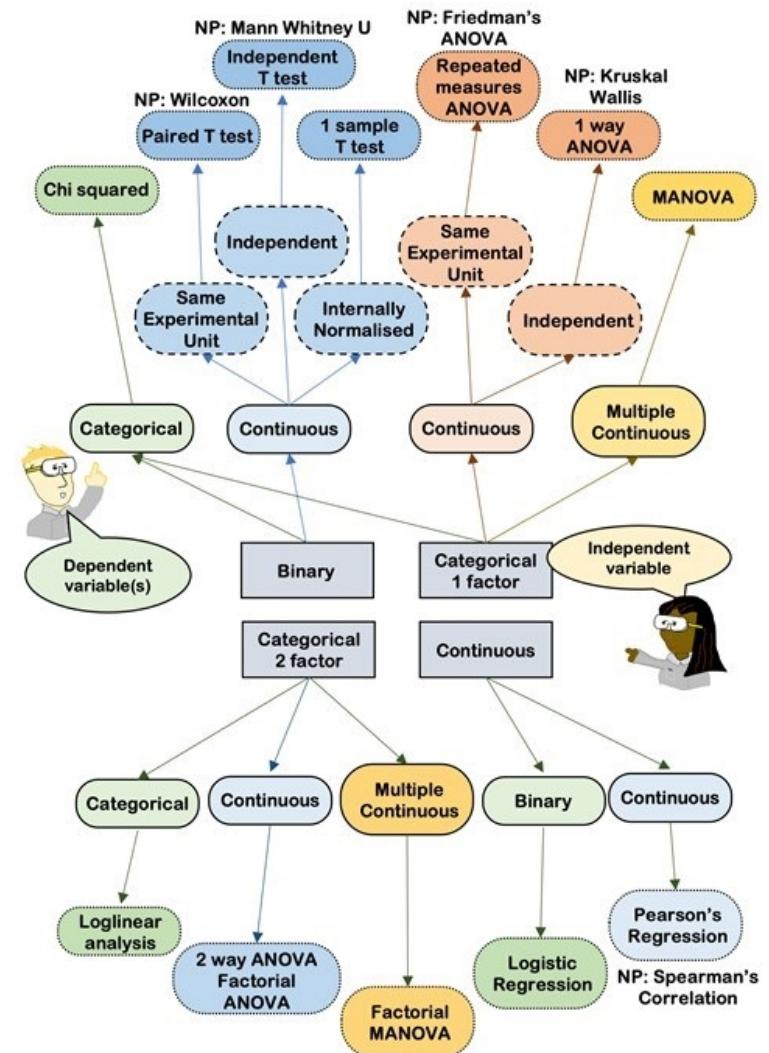
frequently reused. Our aim here is to raise awareness of the problem.

We downloaded and screened supplementary files from 18 journals published between 2005 and 2015 using a suite of shell scripts. Excel files (.xls and.xlsx suffixes) were converted to tabular separated files (tsv) with ssconvert (v1.12.9). Each sheet within the Excel file was converted to a separate tsv file. Each column of data in the tsv file was screened for the presence of gene symbols. If the first 20 rows of a column contained five or more gene symbols, then it was suspected to be a list of gene symbols, and then a regular expression (regex) search of the entire column was applied to identify gene symbol errors. Official gene symbols from Ensembl version 82, accessed November 2015, were obtained for

# Módulo 1.4. Distribuciones de probabilidad y estadística básica

# Nota sobre la taxonomía o nomenclatura de test

- Muchos nombres para muchos test
- Nomenclatura variable según autores, corrientes, épocas
- Dificulta la comprensión o el razonamiento



# Tipos de variables

- Cuantitativas (se expresa mediante numeros, se puede operar)
  - Continuas (decimales)
  - Discretas (enteros)

# Tipos de variables

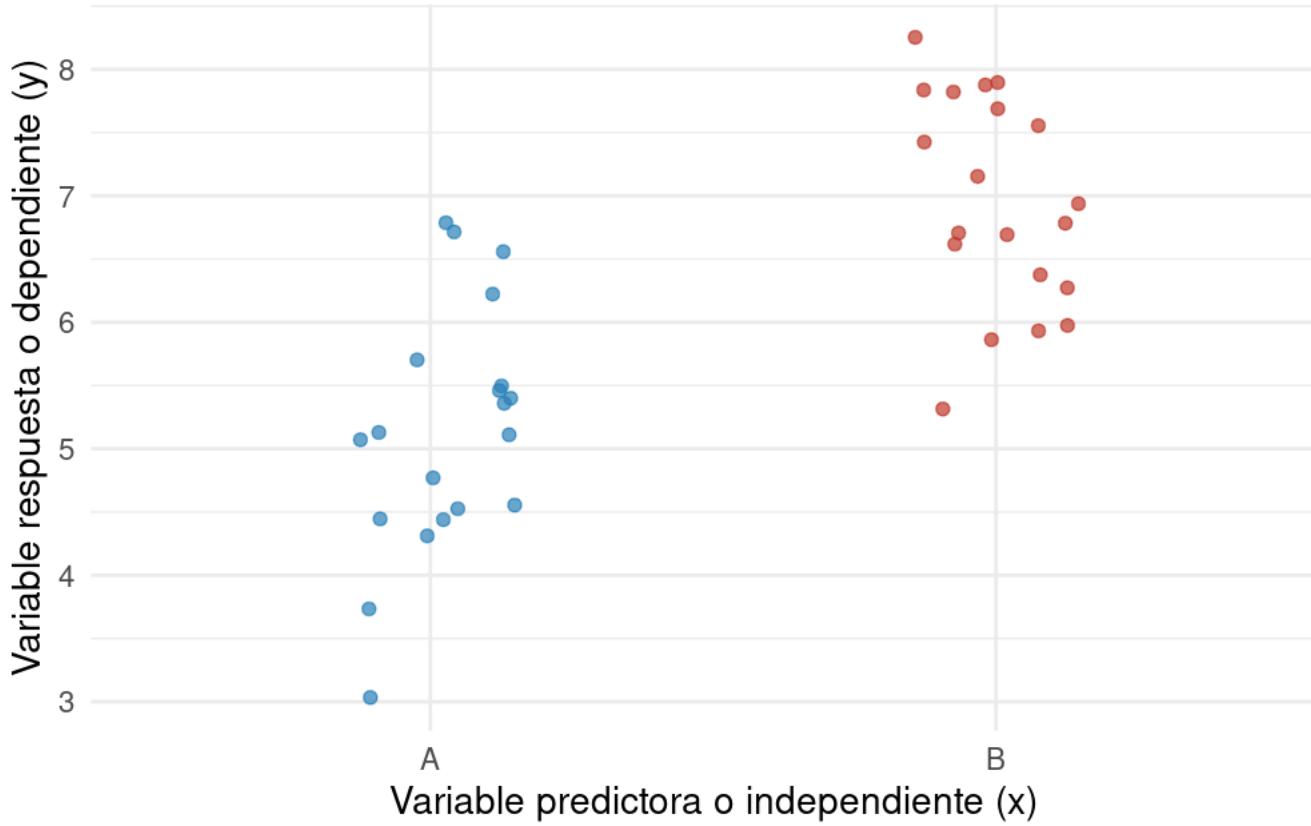
- Cuantitativas (se expresa mediante numeros, se puede operar)
  - Continuas (decimales)
  - Discretas (enteros)
- Semicuantitativas u ordinales (variable cualitativa que se puede ordenar)

# Tipos de variables

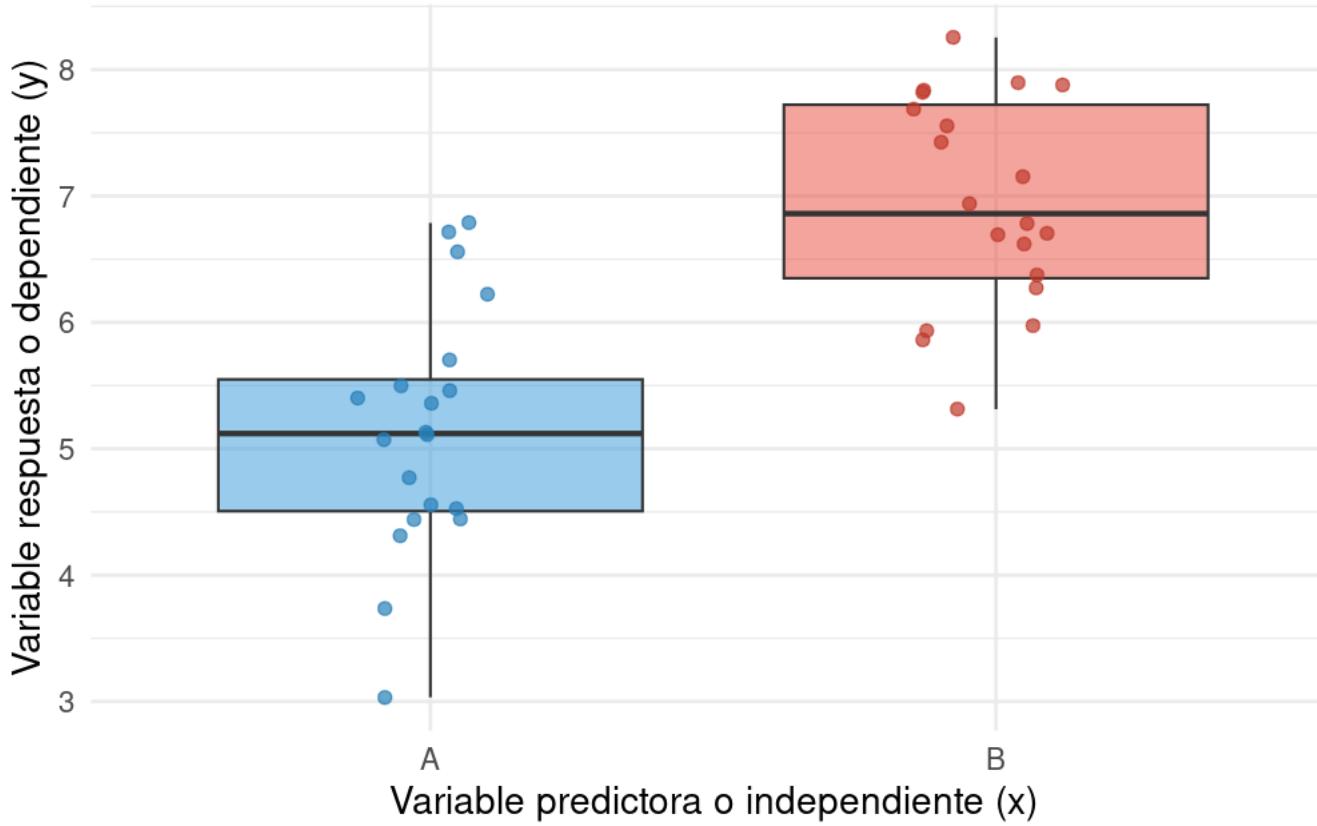
- Cuantitativas (se expresa mediante numeros, se puede operar)
  - Continuas (decimales)
  - Discretas (enteros)
- Semicuantitativas u ordinales (variable cualitativa que se puede ordenar)
- Cualitativas o nominales (describe un atributo o categoría)

# Comparación entre medias

## Comparación de medias entre dos grupos



## Comparación de medias entre dos grupos



# Comparación entre medias

- T-test o t de Student
- ANOVA

# Comparación entre medias: algunos conceptos

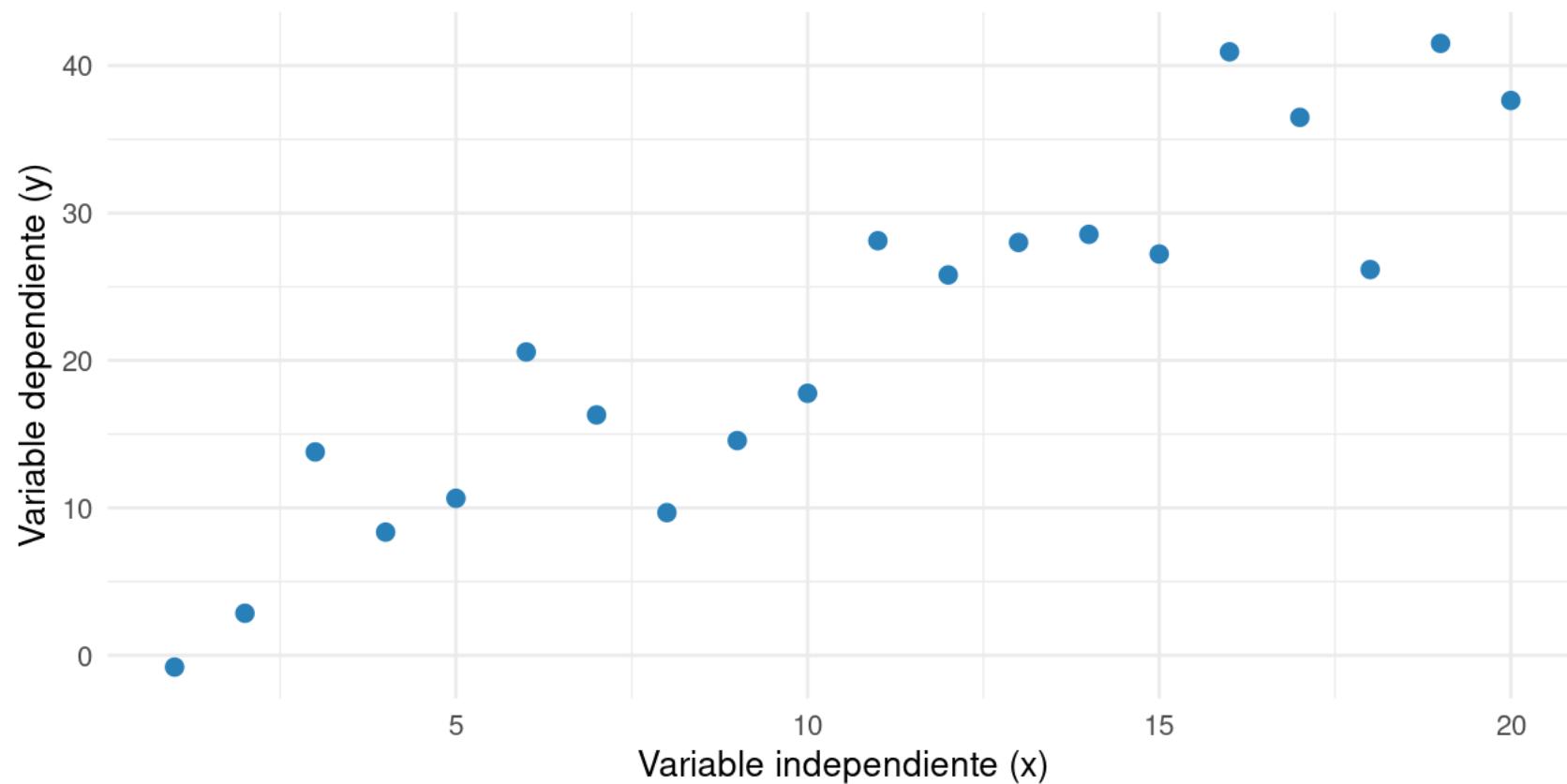
- Variables dependientes o respuesta
- Variables independientes o predictores
- P-valor o significación
- Tipos de errores

# Comparación entre dos variables cuantitativas continuas

- Correlación de Pearson
- Regresión



● Datos observados



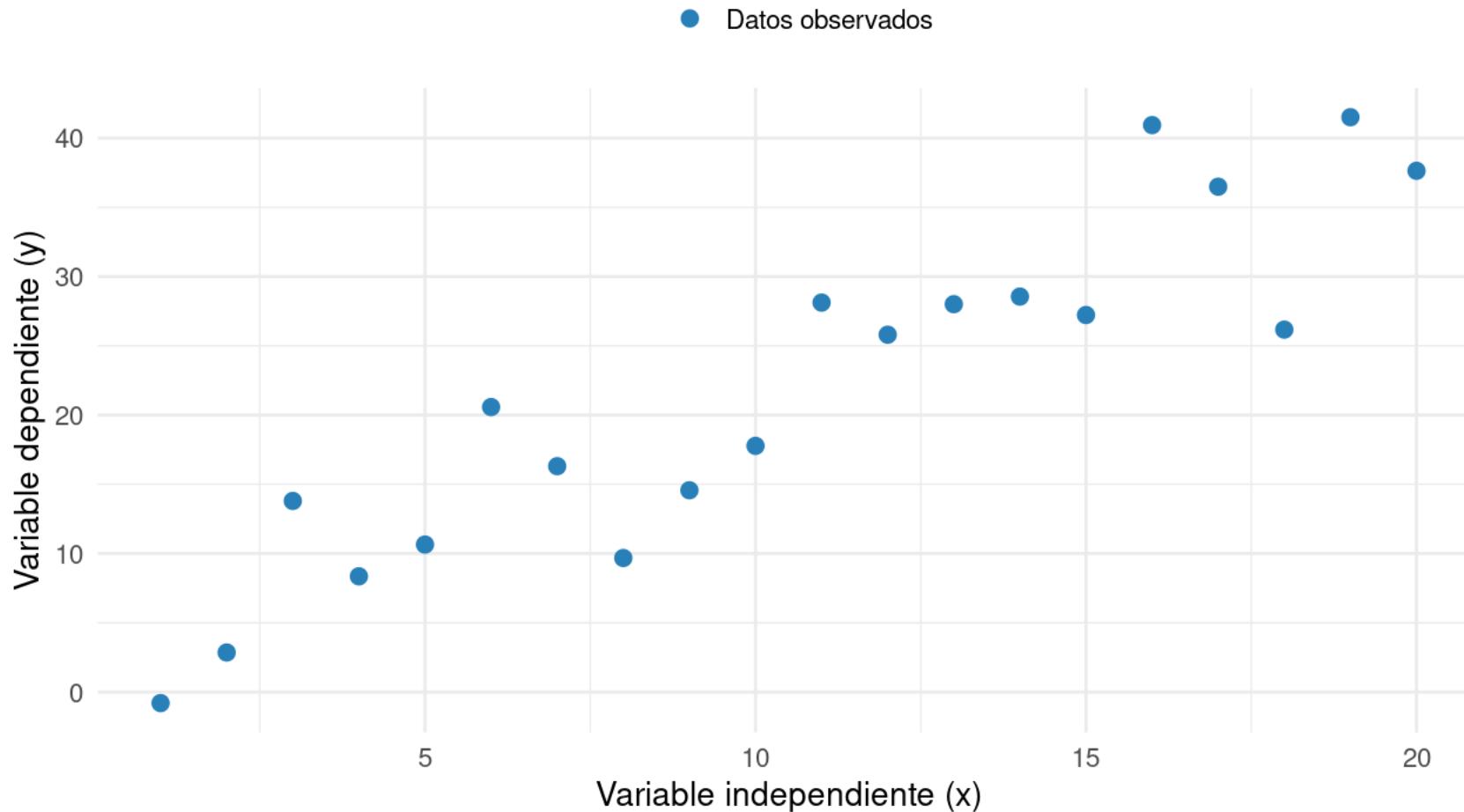
# Comparación entre dos variables cuantitativas continuas

- Correlación de Pearson
- Regresión

# Concepto de residuo

## Visualización de residuos en un modelo lineal

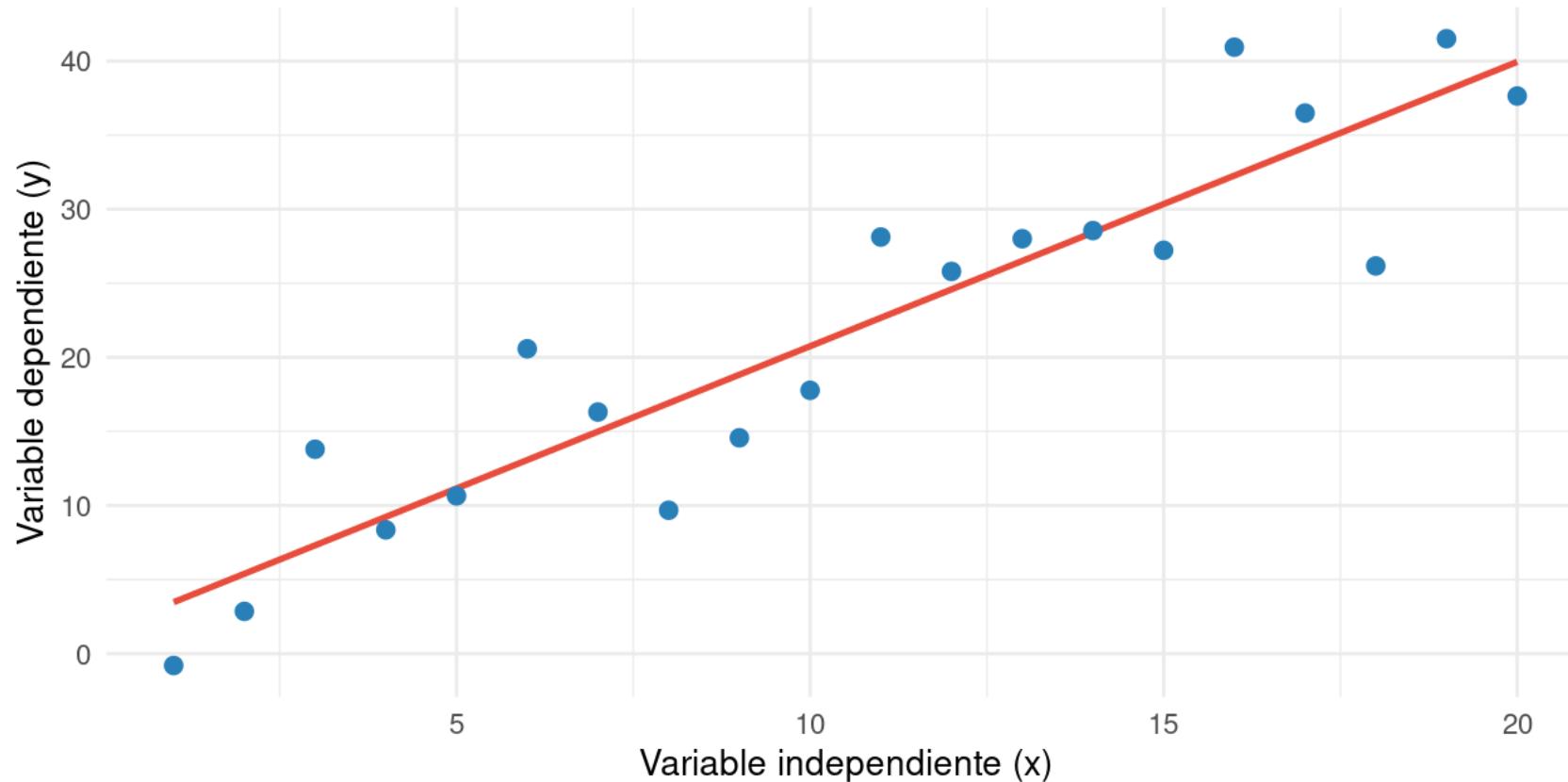
Líneas verdes punteadas indican residuos (diferencias entre valor observado y predicho)



## Visualización de residuos en un modelo lineal

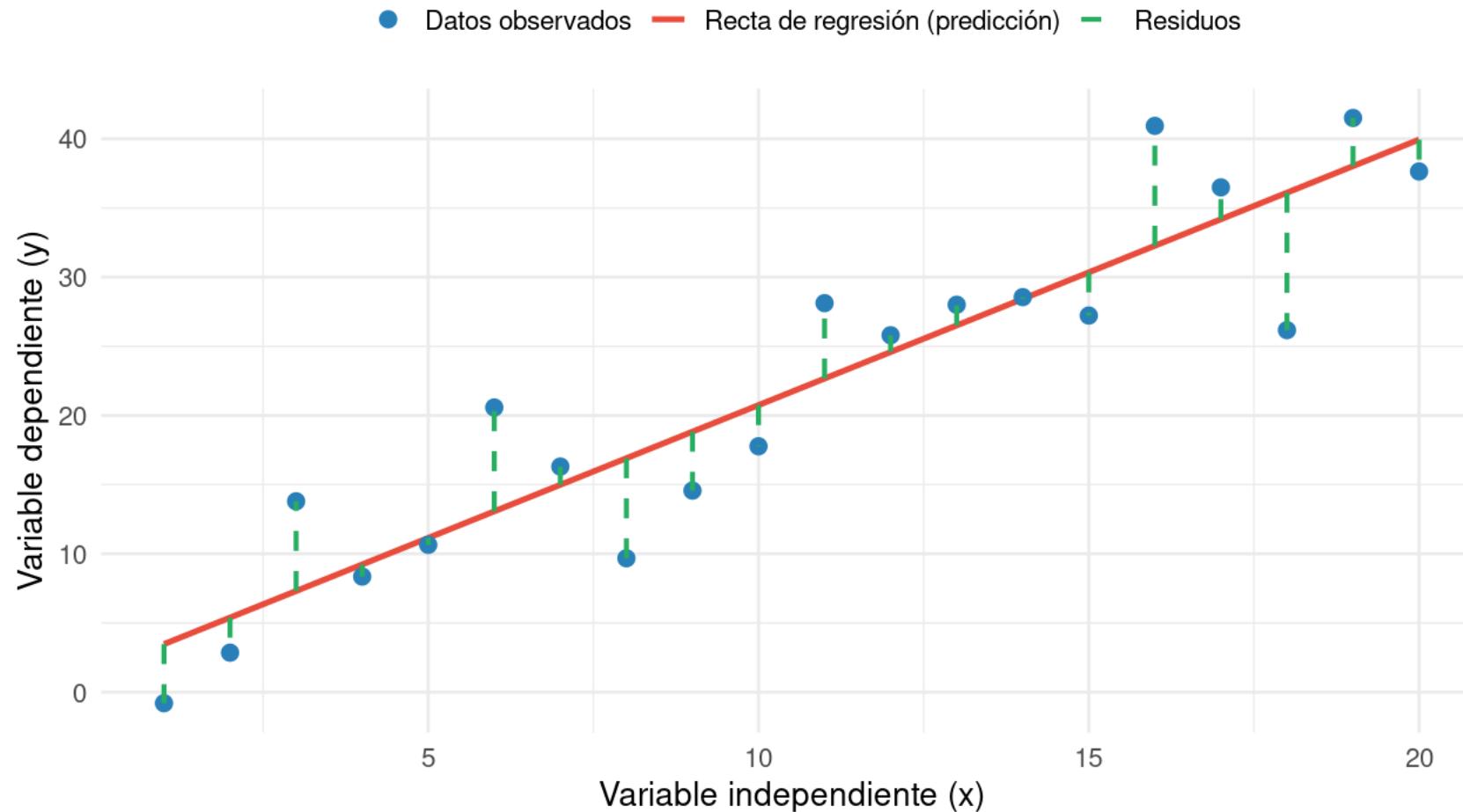
Líneas verdes punteadas indican residuos (diferencias entre valor observado y predicho)

● Datos observados — Recta de regresión (predicción)

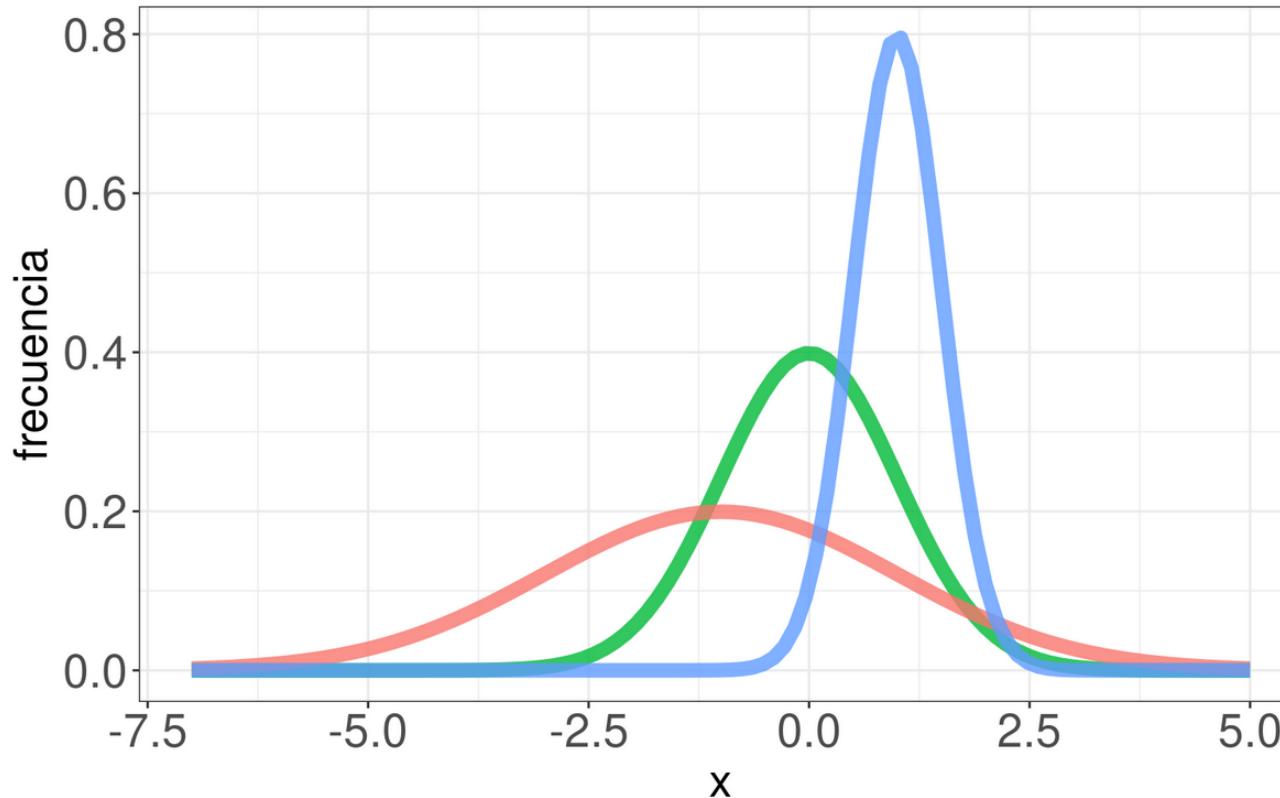


## Visualización de residuos en un modelo lineal

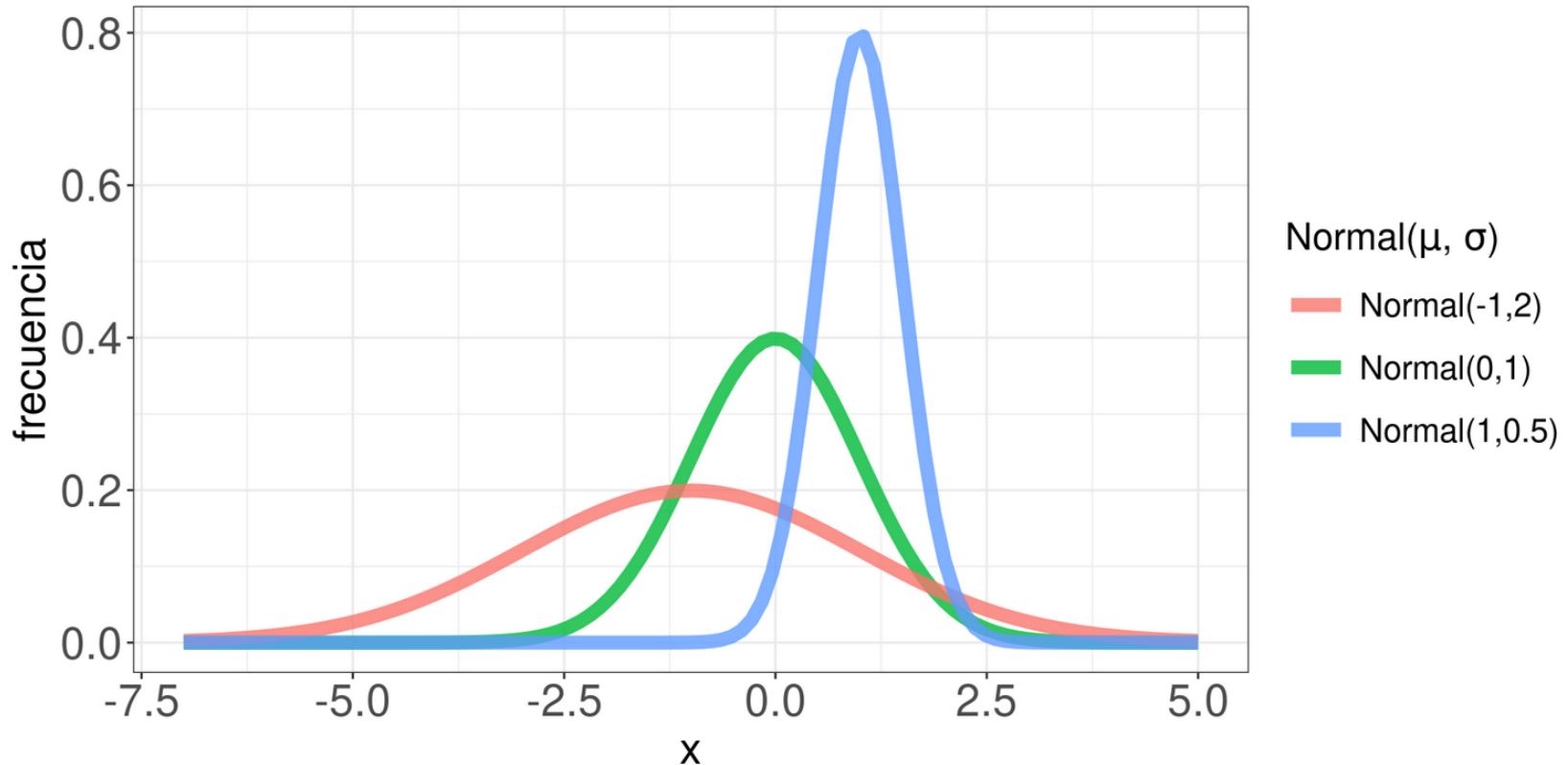
Líneas verdes punteadas indican residuos (diferencias entre valor observado y predicho)



# Distribuciones de probabilidad: distribución normal o Gaussiana



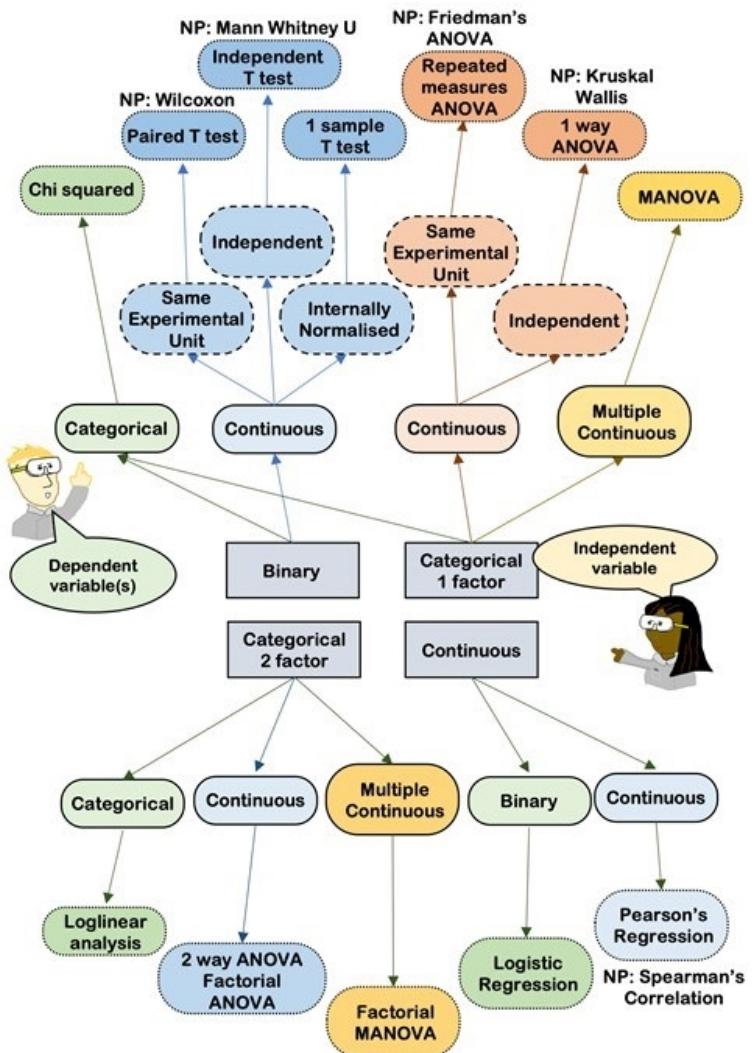
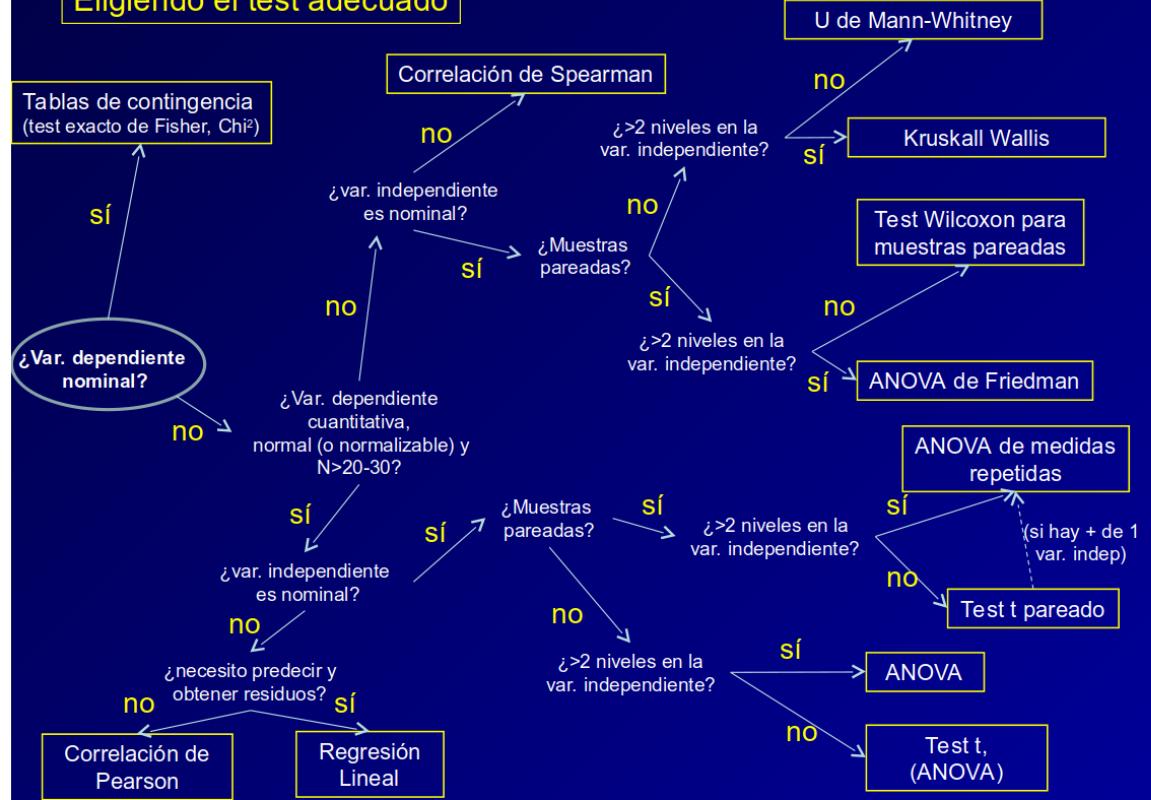
# Distribuciones de probabilidad: distribución normal o Gaussiana



# Distribuciones de probabilidad: distribución normal o Gaussiana

- Shapiro-Wilk normality test
- Q-Q plot

## Eligiendo el test adecuado



## Eligiendo el test adecuado

Tablas de contingencia  
(test exacto de Fisher,  $\chi^2$ )

¿Var. dependiente nominal?  
sí → Tablas de contingencia  
no → ¿Var. cuantitativa normal (o rango) N>20?

¿Var. independiente es nominal?  
no → ¿necesito predecir y obtener residuos?

Correlación de Pearson

Correlación

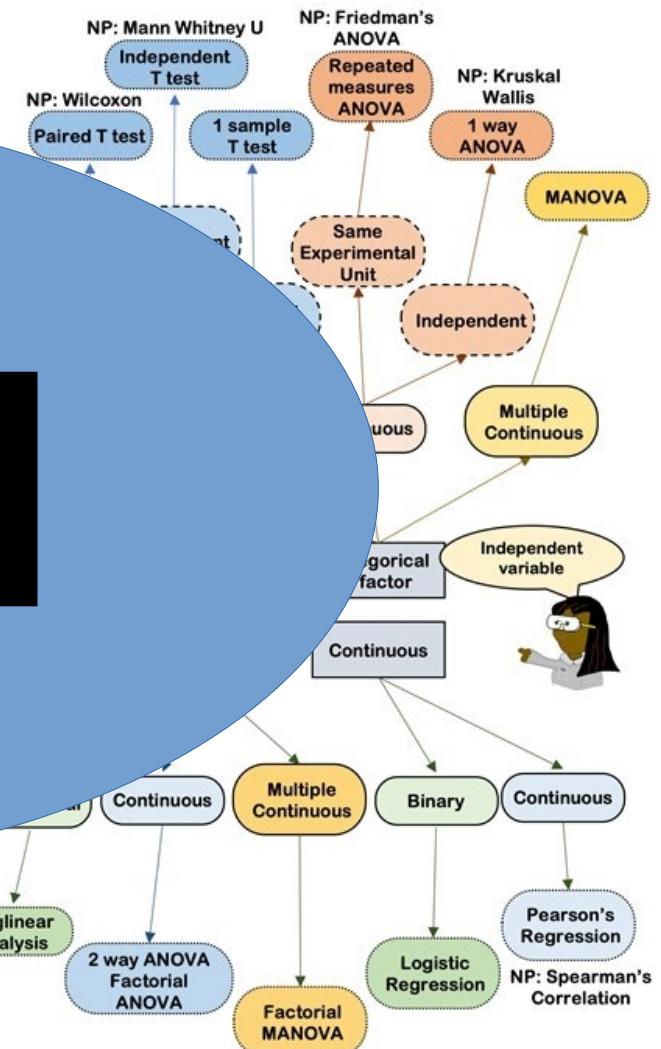
¿var.

¿>2 niveles var. independiente?

Test t,  
(ANOVA)

Regresión Lineal

# GLM



# Common statistical tests are linear models

Last updated: 02 April, 2019

See worked examples and more details at the accompanying notebook: <https://lindelov.github.io/tests-as-linear>

Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon	
<b>Simple regression: <math>\text{Im}(y \sim 1 + x)</math></b>	<b>y is independent of x</b> P: One-sample t-test N: Wilcoxon signed-rank	<code>t.test(y)</code> <code>wilcox.test(y)</code>	<code>lm(y ~ 1)</code> <code>lm(signed_rank(y) ~ 1)</code>	✓ <a href="#">for N &gt; 14</a>	One number (intercept, i.e., the mean) predicts <b>y</b> . - (Same, but it predicts the <i>signed rank</i> of <b>y</b> .)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	<code>t.test(y1, y2, paired=TRUE)</code> <code>wilcox.test(y1, y2, paired=TRUE)</code>	<code>lm(y2 - y1 ~ 1)</code> <code>lm(signed_rank(y2 - y1) ~ 1)</code>	✓ <a href="#">for N &gt; 14</a>	One intercept predicts the pairwise <b>y<sub>2</sub>-y<sub>1</sub></b> differences. - (Same, but it predicts the <i>signed rank</i> of <b>y<sub>2</sub>-y<sub>1</sub></b> .)	
	<b>y ~ continuous x</b> P: Pearson correlation N: Spearman correlation	<code>cor.test(x, y, method='Pearson')</code> <code>cor.test(x, y, method='Spearman')</code>	<code>lm(y ~ 1 + x)</code> <code>lm(rank(y) ~ 1 + rank(x))</code>	✓ <a href="#">for N &gt; 10</a>	One intercept plus <b>x</b> multiplied by a number (slope) predicts <b>y</b> . - (Same, but with <i>ranked</i> <b>x</b> and <b>y</b> )	
	<b>y ~ discrete x</b> P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	<code>t.test(y1, y2, var.equal=TRUE)</code> <code>t.test(y1, y2, var.equal=FALSE)</code> <code>wilcox.test(y1, y2)</code>	<code>lm(y ~ 1 + G<sub>2</sub>)<sup>A</sup></code> <code>glm(y ~ 1 + G<sub>2</sub>, weights=...)<sup>A</sup></code> <code>lm(signed_rank(y) ~ 1 + G<sub>2</sub>)<sup>A</sup></code>	✓ ✓ <a href="#">for N &gt; 11</a>	An intercept for <b>group 1</b> (plus a difference if <b>group 2</b> ) predicts <b>y</b> . - (Same, but with one variance <i>per group</i> instead of one common.) - (Same, but it predicts the <i>signed rank</i> of <b>y</b> .)	
<b>Multiple regression: <math>\text{Im}(y \sim 1 + x_1 + x_2 + \dots)</math></b>	P: One-way ANOVA N: Kruskal-Wallis	<code>aov(y ~ group)</code> <code>kruskal.test(y ~ group)</code>	<code>lm(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub>)<sup>A</sup></code> <code>lm(rank(y) ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub>)<sup>A</sup></code>	✓ <a href="#">for N &gt; 11</a>	An intercept for <b>group 1</b> (plus a difference if group ≠ 1) predicts <b>y</b> . - (Same, but it predicts the <i>rank</i> of <b>y</b> .)	
	P: One-way ANCOVA	<code>aov(y ~ group + x)</code>	<code>lm(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub> + x)<sup>A</sup></code>	✓	- (Same, but plus a slope on <b>x</b> .) <i>Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.</i>	
	P: Two-way ANOVA	<code>aov(y ~ group * sex)</code>	<code>lm(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub> + S<sub>2</sub> + S<sub>3</sub> + ... + S<sub>K</sub> + G<sub>2</sub>*S<sub>2</sub>+G<sub>3</sub>*S<sub>3</sub>...+G<sub>N</sub>*S<sub>K</sub>)</code>	✓	Interaction term: changing <b>sex</b> changes the <b>y ~ group</b> parameters. <i>Note: G<sub>2 to N</sub> is an indicator (0 or 1) for each non-intercept levels of the <b>group</b> variable. Similarly for S<sub>2 to K</sub> for sex. The first line (with G<sub>i</sub>) is main effect of group, the second (with S<sub>j</sub>) for sex and the third is the <b>group × sex</b> interaction. For two levels (e.g. male/female), line 2 would just be "S<sub>2</sub>" and line 3 would be S<sub>2</sub> multiplied with each G<sub>i</sub>.</i>	[Coming]
	<b>Counts ~ discrete x</b> N: Chi-square test	<code>chisq.test(groupXsex_table)</code>	<b>Equivalent log-linear model</b> <code>glm(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub> + S<sub>2</sub> + S<sub>3</sub> + ... + S<sub>K</sub> + G<sub>2</sub>*S<sub>2</sub>+G<sub>3</sub>*S<sub>3</sub>...+G<sub>N</sub>*S<sub>K</sub>, family=...)<sup>A</sup></code>	✓	Interaction term: (Same as Two-way ANOVA.) <i>Note: Run glm using the following arguments: glm(model, family=poisson()) As linear-model, the Chi-square test is log(y) = log(N) + log(a) + log(b) + log(aβ) where a and b are proportions. See more info in the accompanying notebook.</i>	Same as Two-way ANOVA
N: Goodness of fit	<code>chisq.test(y)</code>	<code>glm(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub>, family=...)<sup>A</sup></code>	✓	(Same as One-way ANOVA and see Chi-Square note.)	1W-ANOVA	

List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation  $y \sim 1 + x$  is R shorthand for  $y = 1 \cdot b + a \cdot x$  which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they *all* are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see "Exact" column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `signed_rank = function(x) sign(x) * rank(abs(x))`. The variables G<sub>i</sub> and S<sub>j</sub> are *dummy coded* indicator variables (either 0 or 1) exploiting the fact that when  $\Delta x = 1$  between categories the difference equals the slope. Subscripts (e.g., G<sub>2</sub> or y<sub>1</sub>) indicate different columns in data. lm requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at <https://lindelov.github.io/tests-as-linear>.

<sup>A</sup> See the note to the two-way ANOVA for explanation of the notation.

<sup>B</sup> Same model, but with one variance per group: `glm(value ~ 1 + G2, weights = varIdent(form = ~1|group), method="ML")`.



# Generalized linear model

**Respuesta ~ predictores**

## Take-home messages

- Es importante reflexionar sobre el tipo de nuestras variables
- Necesitamos identificar nuestras variables respuesta y nuestros predictores
- Pensar en modelos lineales puede facilitar el aprendizaje

# **Módulo 2. Test estadísticos en ciencias experimentales (11, 13 y 16 de junio)**

1. Distribuciones de probabilidad y estadística básica
2. Principales test paramétricos y no paramétricos
3. El modelo lineal generalizado
4. Diseños avanzados: medidas repetidas